



# **HUOLTOSEURANTAJÄRJESTELMÄ**

Oskari Laaksonen

Juho Raevaara

Opinnäytetyö  
Toukokuu 2015  
Tietotekniikan Ko  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikan suuntautumisvaihtoehto

LAAKSONEN, OSKARI & RAEVAARA, JUHO:  
Huoltoseurantajärjestelmä

Opinnäytetyö 43 sivua, joista liitteitä 4 sivua  
Toukokuu 2015

---

Huoltoseurantajärjestelmä on asiakkaan toimeksiannosta kehitetty ohjelmisto teollisuudessa toimivien yritysten laitteiden huoltoon ja kunnossapitoon. Järjestelmää ohjataan verkkoselaimella toimivan käyttöliittymän kautta ja kaikki tiedot tallennetaan tietokantaan. Tavoitteena oli luoda ohjelmisto, joka on helppokäyttöinen ja jolla voidaan korvata aiemmin käytetyt menetelmät laitteiden ja huoltojen seurannassa.

Huoltoseurantajärjestelmä toimii Linux-palvelimella ja se on toteutettu tunnettuja ohjelmointikieliä käyttäen. Tämä helpottaa järjestelmän ylläpitoa ja jatkokehitystä, joka tapahtuu Tampereen ammattikorkeakoulun opiskelijoiden toimesta. Tässä opinnäytetyössä esitellään järjestelmän ensimmäistä versiota.

---

Asiasanat: tietojärjestelmät, tietokannat, PHP, MySQL, JavaScript

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Information Technology  
Software Engineering

LAAKSONEN, OSKARI & RAEVAARA, JUHO:  
Maintenance monitoring system

Bachelor's thesis 43 pages, appendices 4 pages  
May 2015

---

This maintenance monitoring system is a software developed on behalf of the client for industrial companies to monitor equipment servicing and maintenance. The system is controlled through internet browser user interface, and all information is stored in a database. The goal was to create an easy to use software which can replace old methods of equipment servicing and maintenance monitoring.

The system runs on a Linux server and it is done using well known programming languages. This makes it easier to maintain and further develop the system, which is done by the students of Tampere University of Applied Sciences. This thesis introduces the first version of the maintenance monitoring system.

---

Key words: information system, database, PHP, MySQL, JavaScript

## Sisällys

1	JOHDANTO .....	7
2	JÄRJESTELMÄN KUVAUS .....	9
3	TOTEUTUS .....	11
3.1	Arkkitehtuuri .....	11
3.2	Ohjelmointikielet .....	12
3.3	Kirjastot .....	13
3.3.1	Bootstrap .....	13
3.3.2	Tablesorter .....	13
3.4	Työkalut .....	15
3.4.1	Notepad++ .....	15
3.4.2	MySQLWorkbench .....	15
3.4.3	Versionhallinta .....	16
3.5	MySQL .....	16
3.6	Apache2 web-server .....	16
4	HUOLTOSEURANTAJÄRJESTELMÄ .....	18
4.1	Yleistä .....	18
4.2	Sisäänkirjautuminen .....	19
4.3	Uusien käyttäjätunnusten luonti .....	20
4.4	Laitteiden lisääminen .....	20
4.5	Laitteiden tarkastelu .....	21
4.6	Laitekortti .....	22
4.7	Huoltokortti .....	23
4.8	Huomiot .....	23
4.9	Huoltojen haku .....	24
4.10	Dokumentit .....	24
4.11	Yhteystiedot .....	25
4.12	Hallintapaneeli .....	25
4.12.1	Käyttäjien lisäys ja muokkaus .....	25
4.12.2	Dokumenttien lisäys ja muokkaus .....	26
4.12.3	Yhteystietojen lisäys ja muokkaus .....	26
4.13	Huoltotriggeri .....	27
5	KESKEISET OSAT .....	28
5.1	Tiedostojen tallennus .....	28
5.2	Laitteiden lisääminen .....	29
5.3	Hallintapaneeli .....	30
5.4	Sähköpostilista .....	30
5.5	.htaccess .....	31

5.6	Tietokanta .....	32
5.7	Tablesorter .....	33
6	JATKOTOIMENPITEET .....	34
6.1	Ylläpito .....	34
6.2	Asiakkaan käytössä ilmenevät asiat .....	34
6.3	Uusien ominaisuuksien kehitys .....	34
7	YHTEENVETO .....	36
	LÄHTEET .....	38
	LIITTEET .....	40
	Liite 1. Järjestelmän navigoinnin rakenne .....	40
	Liite 2. Käyttötapaus – Laitteen lisäys .....	41
	Liite 3. Käyttötapaus – Huollon lisäys .....	42
	Liite 4. Käyttötapaus – Huollon merkkäminen tehdyksi ja huoltoraportin lisääminen .....	43

**LYHENTEET JA TERMIT**

.htaccess	Hypertext Access, Apache-palvelinohjelman hakemistokoh- tainen asetustiedosto
CSS	Cascading Style Sheets
ER-kaavio	Entity-Relationship -kaavio
GIT	avoimen lähdekoodin versionhallintaohjelmisto
html	HyperText Markup Language
IP-osoite	Internet Protocol-osoite
MEAN-stack	MongoDB:n, Express.js:n, Angular.js:n ja Node.js:n muo- dostama sovelluskehys
MySQL	Relaatiotietokantaohjelmisto
MySQLi	MySQL improved, yksi mahdollinen PHP:n rajapinta tieto- kannan käyttöön
NoSQL	Not only SQL
php	HyperText Preprocessor
PDO	PHP Data Object, yksi mahdollinen PHP:n rajapinta tieto- kannan käyttöön
SHA-512	Secure Hash Algorithm, kryptografinen tiivistefunktio esi- merkiksi salasanojen salaukseen
SQL	Structured Query Language
ssh	Secure Shell, salattuun tietoliikenteeseen tarkoitettu proto- kolla
SVN	Subversion, versionhallintaohjelma
TAMK	Tampereen ammattikorkeakoulu
TKI Verstaas	Tietotekniikan Tuotekehitys ja Innovointi Verstaas

## 1 JOHDANTO

Tässä dokumentissa kerrotaan projektista, jonka tarkoituksena oli luoda asiakkaalle uusi huoltoseurantajärjestelmä. Huoltoseurantajärjestelmä tulisi toimimaan verkkosivujen kautta vaatien sisäänkirjautumista ja tiedot tallennettaisiin tietokantaan. Aikaisempi huoltojen seuranta tapahtui Excel-taulukoiden avulla, jossa jokaisella toimipisteellä oli oma Excel-taulukonsa.

Vanha menetelmä on kankea, eikä se mahdollista tietojen muokkausta tai lisäystä missä tahansa. Excel-taulukkoon tallennetaan jokainen toimipisteen laite omalle rivilleen ja laitteiden perustiedot omiin sarakkeisiinsa. Tietojen muokkausta tai lisäämistä varten pitää aina löytää oikean toimipaikan taulukko, joko muistitikulta, sähköpostista tai paikallisesti tietokoneelta. Kun oikea taulukko on löydetty, pitää sieltä etsiä oikea laite, ilman varsinaisia hakutoimintoja, ja lisätä esimerkiksi huolto oikeaan sarakkeeseen. Tämän lisäksi tiedostojen, esimerkiksi huoltoraporttien, lisääminen halutulle laitteelle myöhempää tarkastelua varten on vaikeaa. Taulukkoon voidaan tallentaa korkeintaan linkki haluttuun tiedostoon. Tällöin lisätty tiedosto pitää siirtää itse Excel-taulukon mukana, mikäli sitä halutaan tutkia myöhemmin. Suoran liiketoimintaan liittyvän Excel-taulukon lähetyks edestakaisin sähköpostin välityksellä tai muistitikulla on myös riskialtista tiedon paikkansapitävyyden kannalta.

Uuden huoltoseurantajärjestelmän on tarkoitus toimia verkkoselaimen kautta. Näin järjestelmän käyttö onnistuu missä tahansa, missä on internet yhteys. Tämä helpottaa Järjestelmään lisätään asiakkaiden toimipisteissä käytettäviä laitteita ja tiedot tallennetaan tietokantaan. Kun laitteelle halutaan lisätä esimerkiksi huoltoraportti, tapahtuu se verkkoselain-käyttöliittymän kautta ja tallennetaan palvelimelle, jossa koko järjestelmä pyörii. Näin myös laitteelle lisätyt tiedostot ovat saatavilla helposti ja ne ovat suoraan yhdistetty oikeaan laitteeseen.

Käyttöliittymästä rakennettiin asiakkaan toiveiden mukaisesti mahdollisimman yksinkertainen ja helppokäyttöinen. Järjestelmä on käytettävissä myös mobiililaitteilla, jotta esimerkiksi huoltomiehet pääsevät tietoihin käsiin kätevästi tabletilla tai kännykällä ilman, että he tarvitsisivat pöytäkonetta tai kannettavaa tietokonetta. Kuitenkin käyttäjälle näytetään kaikki tieto mitä käyttäjän tarvitsee nähdä. Tämän ansiosta laitteiden ja

huoltojen ylläpito ja seuranta on helpompaa, sekä parantaa tietojen luotettavuutta vanhaan järjestelmään verrattuna. Lisäksi pelkällä vilkaisulla on mahdollista nähdä, että kaikki on kunnossa tai että jotain erityistä pitää tehdä.

Tämän dokumentin luvussa kaksi käydään läpi uuden järjestelmän tarkempi määrittely. Kolmannessa luvussa kerrotaan tekniikoista ja välineistä joiden avulla järjestelmä toteutettiin. Neljäs luku käy läpi huoltoseurantajärjestelmän ominaisuudet ja viidennessä luvussa perehdytään tarkemmin järjestelmän keskeisimpien osien toteutukseen. Luvussa kuusi kerrotaan jatkotoimenpiteistä, joita järjestelmälle tehdään ja lopuksi luvussa seitsemän tehdään yhteenveto koko projektista.

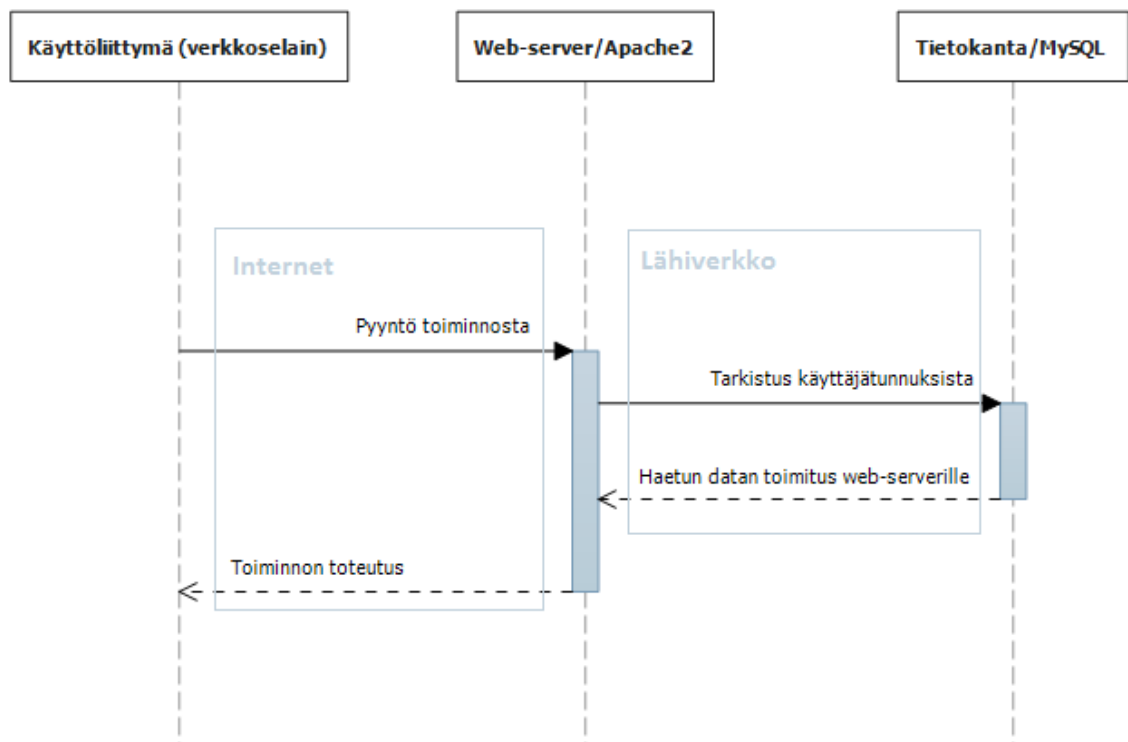


## 2 JÄRJESTELMÄN KUVAUS

Huoltoseurantajärjestelmä suunniteltiin asiakasyritykselle, jonka liiketoiminnasta osa liittyy teollisuudessa käytettävien laitteiden huoltoon ja ylläpitoon. Laitteet, joita järjestelmään tallennetaan, ovat siis teollisuusautomaatioissa käytettäviä laitteita.

Aikaisemmin käytössä oleva toimintatapa pohjautui laitteiden kirjaamisesta Exceliin. Jokaiselle toimipisteelle tehtiin oma tiedosto, johon syötettiin laitteeseen liittyvää dataa. Uudessa järjestelmässä toiminnot kirjataan myös toimipistekohtaisesti, mutta kaikkiin toimipisteisiin pystyy syöttämään yhdellä ylläpitotason sisäänkirjautumisella tietoja, jotka näkyvät eri toimipisteiden henkilöstölle erikseen.

Karkeasti kuvattuna järjestelmän voisi jakaa kolmeen osaan, verkkoselaimessa toimivaan käyttöliittymään, web-serveriin sekä MySQL tietokantaan (kuvio 1). Käyttöliittymässä on kaikki käyttäjille suunnatut elementit, jotka kommunikoivat web-serverin kanssa, josta taas lähetetään pyyntöjä tietokantaan.



KUVIO 1. Järjestelmä kolmeen osaan eriytettynä

Järjestelmän käyttöliittymän visuaalisesta ilmeestä oli asiakkaalla jo ideoita. Järjestelmän etusivusta oli jo laadittu layout-kuva, sekä sivuston yläpalkista, logoista ja palvelun nimestä oli jo tehty suunnitelma. Näitä kuitenkin muutettiin sopivimmiksi järjestelmää kehitettäessä.

Yksi järjestelmän tärkeimmistä osista ovat laitekortit ja niihin liittyvät laitekohtaiset tietokentät. Erilaisia laitteita järjestelmässä on 11kpl joissa jokaisessa on 20–35 tietokenttää. Useilla laitteilla on noin kymmenen samaa kenttää ja loput kentistä ovat uniikkeja laitteesta riippuen. Asiakkaan määrittelydokumentista ilmeni kunkin laitteen tärkeimmät tietokentät, joskin kenttiä lisättiin jälkikäteen järjestelmää testattaessa.

Lisättyjä laitteita pääsee tarkastelemaan laiterekisteristä, joka muodostaa listan lisätyistä laitteista. Tätä toimintoa tarvitaan siihen, että eri laitteista saa yhdellä vilkaisulla kokonaisvaltaisen kuvan. Järjestelmää suunnitellessa painotettiin sitä, että tässä näkymässä olisi vain asiakkaan kannalta kaikista oleelliset tiedot laitteista ja laitteiden tarkemmat tiedot näkyisivät vasta laitteiden laitekortilta.

Toinen oleellinen osa järjestelmää on erilaisten dokumenttien tallennus- ja lataamismahdollisuus. Jokaiselle laitteelle on mahdollista ylläpidon toimesta lisätä laitekohtaisia yleisiä dokumentteja, kuten kytkentäkaavioita, sekä huoltopöytäkirjoja, jotka liittyvät yksittäisen laitteen yksittäiseen huoltoon. Näiden laitekohtaisten dokumenttien lisäksi järjestelmään voi lisätä asiakaskohtaisia yleisiä dokumentteja, esimerkiksi hinnastoja.

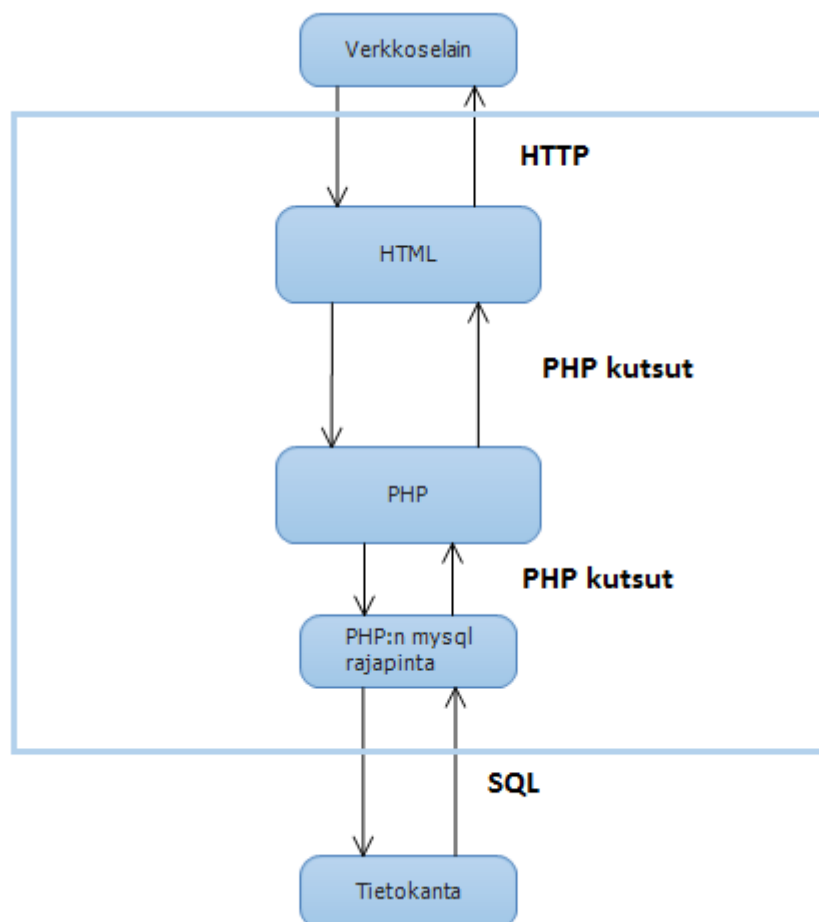
Järjestelmään tallennetaan myös kaikkien tahojen yhteystietoja. Kaikkien asiakkaiden yhteystiedot ovat tarkasteltavissa ja muokattavissa ylläpidon toimesta ja asiakas näkee pelkästään heille tarpeelliset yhteystiedot. Yhteystietoihin tallennetaan asiakkaan nimi, paikkakunta, titteli, sähköposti, puhelinnumero ja tekniikka-alueet.

Käyttöliittymän tulee toimia uusimmilla selaimilla, vähimmäisvaatimuksena ainakin Internet Explorer 10, Google Chrome sekä Mozilla Firefox. Lisäksi järjestelmän haluttiin toimivan mobiililaitteilla. Mobiililaitteille ei tehty erillistä näkymää, mutta järjestelmän ulkoasun toteutustavasta johtuen se on hyvin luettavissa ja käytettävissä ilman erillistä mobiilinäkymää. Mobiilinäkymä on kuitenkin osa mahdollista jatkokehitystä.

### 3 TOTEUTUS

#### 3.1 Arkkitehtuuri

Eri osia järjestelmässä ohjelmistotasolla on kaksi kappaletta, HTML ja PHP. Yksi PHP-tiedosto sisältää yhden näkymän ja sivujen välillä siirtyvä data lähetetään pääosin POST-metodeilla, poikkeuksena hallintapaneeli, jossa näkymien vaihtuminen tapahtuu PHP-funktioilla. HTML-koodi hoitaa kaikki verkkoselaimessa toimivaan käyttöliittymään liittyvät ominaisuudet, kuten sivuston ulkonäön, napit, syöttökentät ja valikot. PHP hoitaa kaikki palvelinpuolella tapahtuvat asiat, kuten tallennettujen tiedostojen hakemistojen tarkistelun ja kaikki SQL-kyselyt (kuvio 2).



KUVIO 2. HTML ja PHP ovat järjestelmän keskeisimmät osat

Joitain osia järjestelmästä on tehty myös pelkästään PHP:llä, kuten tiedostojen lähetys palvelimen oikeisiin hakemistoihin, uusien laitteiden tallennus ja muokkaus, sekä automaattisesti käynnistyvä sähköpostiscripti. Tietojen välitys tiedostojen tallennukseen ja

uusien laitteiden lisäämiseen välitetään HTML:n POST- ja GET-metodeilla. Näitä tiedostoja käsitellään samoin, kuin mitä tahansa muutakin PHP tiedostoa, vaikka tiedostojen tallennukseen ja muokkaukseen sekä tiedostojen lisäämiseen käytettävissä tiedostoissa ei ole mitään näkyvää käyttöliittymäosaa.

PHP-tiedostoja joissa ei ole varsinaista HTML osiota voisi lähestyä myös oliomaisemalla ohjelmointitavalla, jossa tehtäisiin luokat jokaisesta toiminnosta ja niistä luotuja olioita kutsuttaisiin PHP koodista. Tällä voitaisiin myös välttyä ainakin osittain POST- ja GET-metodien käytöstä. Järjestelmän kehitysvaiheessa päädyttiin kuitenkin pitämään tiedostojen käsittelyt sekä laitteiden lisäys erillisissä tiedostoissa pelkkänä perinteisenä PHP koodina, eikä niistä tehty olioita.

Järjestelmässä käytettävään tapaan päädyttiin lähinnä siksi, koska ensimmäisissä versioissa tiedostojen käsittelyt ja laitteiden luonnit tehtiin aina samaan tiedostoon ja näitä tiedostoja karsimalla päädyttiin nykyiseen ratkaisuun, joka osoittautui erittäin toimivaksi juuri tähän käyttötarkoitukseen.

### 3.2 Ohjelmointikielet

Järjestelmän pääohjelmointikielet ovat HTML, PHP ja JavaScript. HTML on web-ohjelmoinnin standardikieli, jolla käytännössä kaikki verkkosivut luodaan. PHP on palvelinpuolen ohjelmointiin käytettävä skriptikieli, jolla esimerkiksi kaikki järjestelmän tietokantakomennot suoritetaan. Kaikki käyttöliittymässä esiintyvä dynaamiset ominaisuudet ovat toteutettu JavaScriptillä. Järjestelmän tietokantana on MySQL jonka ohjaamiseen käytetään SQL –kieltä, jolle on hyvä tuki PHP:ssä.

Kaikki järjestelmässä käytettävät ohjelmointikielet ovat yleisesti erittäin paljon käytettyjä ja joiden kehitys on jatkunut jo useita vuosia. Tämä takaa järjestelmän helpon ylläpidettävyyden ja sen, että järjestelmää voi ajaa melkein millä tahansa palvelinympäristöllä, eikä erikoisia palvelinohjelmistoja vaadita.

Järjestelmän olisi voinut myös toteuttaa uudemmilla web-tekniikoilla. Client puolen olisi voinut toteuttaa esimerkiksi Angular.js:n ([angularjs.org/](http://angularjs.org/), 2015) avulla. Angular.js olisi sopinut client puolen ohjelmointikieleksi hyvin, sillä se on tarkoitettu dynaamisen

tiedon luomiseen ja näyttämiseen verkkosivuilla. Kuitenkin projektille asetetun tiukan aikataulun ja hyvin vähäisen kokemuksen takia Angular.js:n opettelu ja sen käyttäminen järjestelmän toteutuksessa hylättiin. Palvelin puolen olisi voinut toteuttaa monellakin eri kielellä, ehkä vahvimpana vaihtoehtona oli kuitenkin Node.js ([nodejs.org/](http://nodejs.org/), 2015), josta oli jo hivenen kokemusta.

Lopulta järjestelmä päädyttiin kuitenkin kehittämään perinteisemmillä tavoilla, silmäläpityä tulevaa ylläpitoa ja jatkokehitystä, joka tapahtuu todennäköisesti opiskelijaprojektina. Mikäli järjestelmä olisi kehitetty käyttäen jotain uudempaa tekniikkaa, kuten Angular.js:ää, Node.js:ää, tai vaikkapa näiden yhdistelmää eli MEAN-stackia ([mean.io/](http://mean.io/), 2015), joutuisi ylläpitoon valittu opiskelija opiskelemaan omatoimisesti ylimääräistä, koska opintoihin kuuluu pakollisena vain perinteiset tekniikat.

### **3.3 Kirjastot**

#### **3.3.1 Bootstrap**

Bootstrap on kaikista käytetyin HTML, CSS ja JavaScript kehikko responsiivisten verkkosivuprojektien tekemiseen ([getbootstrap.com](http://getbootstrap.com), 2015). Järjestelmässä käytetään paljon erilaisia nappeja sekä taulukoita, joiden toimintaa ja ulkoasua on helppo muuttaa Bootstrapillä. Bootstrap on julkaistu MIT-lisenssillä, joka antaa luvan ohjelmiston ilmaiseen käyttöön([github.com/twbs/bootstrap/blob/master/LICENSE](https://github.com/twbs/bootstrap/blob/master/LICENSE), 2015).

Bootstrap päädyttiin ottamaan käyttöön sen suuren suosion ja kollegoilta saadun hyvän palautteen ja suosittelun takia. Muita vartenotettavia kehyksiä olisi ollut Foundation ([foundation.zurb.com](http://foundation.zurb.com), 2015) tai hieman kevyempi Gumby ([gumbyframework.com](http://gumbyframework.com), 2015). Kaikilla näillä kehyksillä saisi toteutettua kaikki tässä järjestelmässä käytettävät toiminnot.

#### **3.3.2 Tablesorter**

Tablesorter on jQuery lisäosa, joka auttaa kaikkien HTML taulukoiden järjestelyssä ilman sivun uudelleenlatausta ([tablesorter.com/docs/](http://tablesorter.com/docs/), 2015). Järjestelmän käytetyimmät

osat ovat näkymiä, jotka on toteutettu käyttäen HTML taulukoita (table). Tämä lisäosa tekee mahdolliseksi taulukoiden uudelleenjärjestelyn aakkosjärjestykseen klikkaamalla pystyrivien otsikoita, ilman että koko sivu pitäisi ladata uudelleen.

Tarve taulukoiden järjestelyyn tuli esiin ensimmäisten asiakastapaamisten aikana, jonka jälkeen alettiin miettimään parasta tapaa ratkaista ongelma taulukoiden uudelleen järjestelyyn. Yhtenä vaihtoehtona oli aluksi suorittaa uusia SQL kyselyjä käyttäjän asettamien asetusten perusteella, joka lataisi sivun uudestaan niin, että SQL lauseella olisi vaihdettu priorisointi esimerkiksi jonkun pystysarakkeen aakkosjärjestyksen mukaan. Tämä tapa olisi kuitenkin ollut turhan raskas ja käyttökokemus ei enää olisi ollut sulava.

Tablesorterin käyttöön päädyttiin kuitenkin melko nopeasti, kun asiaa lähdettiin tutkimaan internetin hakukoneista. Monella keskustelupalstalla oli erilaisissa muissa projekteissa ollut samoja tarpeita taulukoiden järjestelyyn ja poikkeuksetta kaikkiin keskusteluihin oli tarjottu yhdeksi vaihtoehdoksi Tablesorteria. Tablesorterilla pystyisi myös muokkaamaan taulukoiden ulkoasua, mutta tätä ominaisuutta ei kuitenkaan käytetty, sillä Bootstrapissa on käyttöliittymän muihin osiin paremmin sopiva taulukkoulkoasu.

### 3.4 Työkalut

#### 3.4.1 Notepad++

Järjestelmän kehitys tehtiin pääosin käyttäen Notepad++ nimistä tekstieditoria ([notepad-plus-plus.org/](http://notepad-plus-plus.org/), 2015). Notepad++ mahdollistaa useamman eri tiedoston avaamisen yhteen ikkunaan, monelle välilehdelle. Ohjelmassa on myös erittäin tehokas hakufunktio, sekä tuki kaikkiin tässä projektissa käytettyihin ohjelmointikieliin.

Tekstieditorin valinta ei vaikuta projektin lopputulokseen juuri mitenkään. Tätä editoria päädyttiin käyttämään, koska se on ilmaiseksi saatavilla ja laajasti käytetty, joten editoriin liittyvissä ongelmatilanteissa on helppo löytää ratkaisu internetin hakukoneista. Muita mahdollisia laajasti käytettyjä editoreja harkittiin myös, esimerkiksi Sublime Text sopii loistavasti verkkosivujen kehitykseen, mutta sen käytöstä joutuu maksamaan 70 euron arvoisen lisenssimaksun ([sublimetext.com/](http://sublimetext.com/), 2015).

#### 3.4.2 MySQLWorkbench

Järjestelmän tietokannaksi valittua MySQL tietokantaa rakennettiin ja muokattiin MySQLWorkbench ([mysql.com/products/workbench/](http://mysql.com/products/workbench/), 2015) -ohjelmistolla, joka on visuaalinen tietokannan suunnittelu-, ylläpito- ja kehitystyökalu. MySQLWorkbenchin avulla on mahdollista muun muassa luoda tietokannan ER-kaavio tyhjästä, rakentaa olemassa olevasta tietokannasta ER-kaavio, tehdä tietokannasta varmuuskopio, jonka avulla samat tiedot ja taulut voi esimerkiksi siirtää uudelle palvelimelle, ja suorittaa normaaleja MySQL tietokantakyselyjä.

MySQLWorkbenchin sijaan olisi voitu käyttää esimerkiksi phpMyAdmin ([phpmyadmin.net/home\\_page/](http://phpmyadmin.net/home_page/), 2015) nimistä ohjelmistoa, joka tarjoaa web-käyttöliittymän sekä työkalut tietokannan suunnitteluun ja ylläpitoon. MySQLWorkbenchin käyttöön päädyttiin kuitenkin siksi, että se tarjoaa visuaalisen esityksen tietokannasta ja sen käytöstä oli jo aikaisempaa kokemusta.

### 3.4.3 Versionhallinta

Projektin kehitysvaiheessa on käytetty versionhallintaan TAMKin tarjoamaa SVN-repositoryä sekä TortoiseSVN ([tortoisesvn.net](http://tortoisesvn.net), 2015) ohjelmistoa. Versionhallintaa käytetään yleisesti kaikissa ohjelmistoprojekteissa, varsinkin jos samassa projektissa on useampia tekijöitä. SVN:ää päädyttiin käyttämään tässä projektissa erittäin ahkerasti, jokaisena kehityspäivänä repositoryyn lähetettiin useampi versio.

Toinen vaihtoehto SVN:lle olisi GIT ([git-scm.com](http://git-scm.com), 2015), joka on saavuttanut suuren suosion versionhallinnassa. SVN:n käyttöön päädyttiin kuitenkin aikaisemman kokemuksen perusteella, sekä siksi, että TAMK tarjoaa repositoryn siihen. Versionhallinta ei ole suoraan yhteydessä varsinaisella web-serverillä olevaan käyttöliittymäversioon, vaan uudet versiot ajetaan palvelimelle manuaalisesti ssh-yhteydellä.

## 3.5 MySQL

MySQL on avoimen lähdekoodin relaatiotietokantaohjelmisto, joka tarjoaa nopean, luotettavan ja helppokäyttöisen tietokannan käytettäväksi erilaisissa järjestelmissä. Se on käytetyin avoimen lähdekoodin tietokantaohjelmisto, jota kehittää ja ylläpitää Oracle Corporation ([dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html](http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html), 2015).

MySQL:n valintaan järjestelmässä käytettäväksi tietokannaksi johti siitä jo omattu kokemus sekä laaja ja pitkälle viedyt PHP kirjastot tietokannan käyttöön. Valittu PHP kirjasto on nimeltään MySQLi, joka on paranneltu versio vanhasta kirjastosta. Vaihtoehtona olisi voitu käyttää vaikkapa PDO-kirjastoa, mutta MySQLi oli entuudestaan tuttu ja sen käytöstä löytyi parhaat ohjeet ja manuaalit. Lisäksi TAMK tarjoaa jokaiselle opiskelijalle opiskelun ajaksi MySQL tietokannan omaan käyttöön, jota hyödynnettiinkin kehitysvaiheessa paljon.

## 3.6 Apache2 web-server

Apache2 on avoimen lähdekoodin palvelinohjelmistoprojekti, jonka tarkoituksena on tarjota turvallinen, tehokas ja laajennettava palvelin, joka tarjoaa HTTP palveluja ajan



tasalla HTTP standardien mukaan([httpd.apache.org](http://httpd.apache.org), 2015). Apachea on käytetty järjestelmän kehitysvaiheessa Windows käyttöjärjestelmällä, sekä sitä käytetään myös varsinaisella Linux-pohjaisella palvelimella.

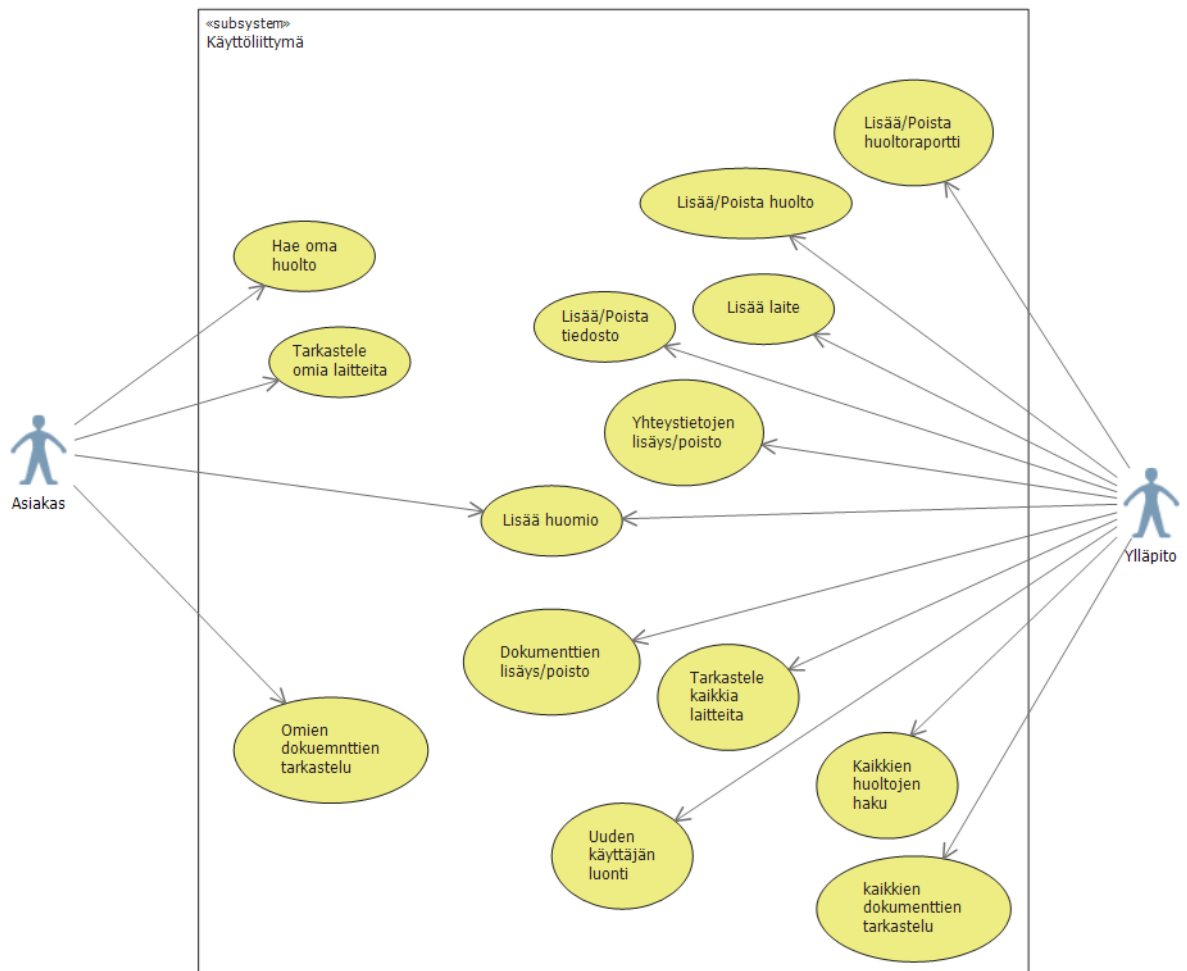
Järjestelmän verkkosivujen kautta toimiva käyttöliittymä on melko kevyt, eikä siinä ole erikoisia ominaisuuksia johon tarvitsisi välttämättä juuri Apachen palvelinohjelmiston. Vartenotettava toinen vaihtoehto olisi ollut esimerkiksi Lighttpd ([lighttpd.net/](http://lighttpd.net/), 2015). Apache on kuitenkin ollut käytössä aikaisemmissa projekteissa ja sen käyttöä on käyty läpi myös koulussa, joten Apache2 on tähän projektiin varmasti paras ratkaisu.

## 4 HUOLTOSEURANTAJÄRJESTELMÄ

### 4.1 Yleistä

Huoltoseurantajärjestelmä toteutettiin käyttäen yksinkertaista ja helposti luettavaa ohjelmakoodia. Tarkoituksena oli kehittää järjestelmä, jonka ylläpito ja jatkokehitys eivät vaadi suurta ohjelmointitaitoa ja kokemusta, pelkät perustiedot PHP:stä, MySQL:stä ja JavaScriptistä riittävät. Lisäksi järjestelmä räätälöitiin täysin asiakkaan tarpeisiin, eikä sitä ole tarkoitus ottaa käyttöön muualla.

Asiakkaan pyynnöstä järjestelmään luotiin eri näkymät ja toiminnollisuudet kahdelle eri tason käyttäjätunnukselle. Asiakkaan käyttäjätunnukset ovat aina korkeamman tason käyttäjätunnuksia, joilla voi käyttää kaikkia järjestelmässä olevia ominaisuuksia. Järjestelmän tilanteen asiakkaan asiakkaat saavat alemman tason käyttäjätunnukset, joiden oikeudet järjestelmässä ovat rajoitetut (kuvio 3).



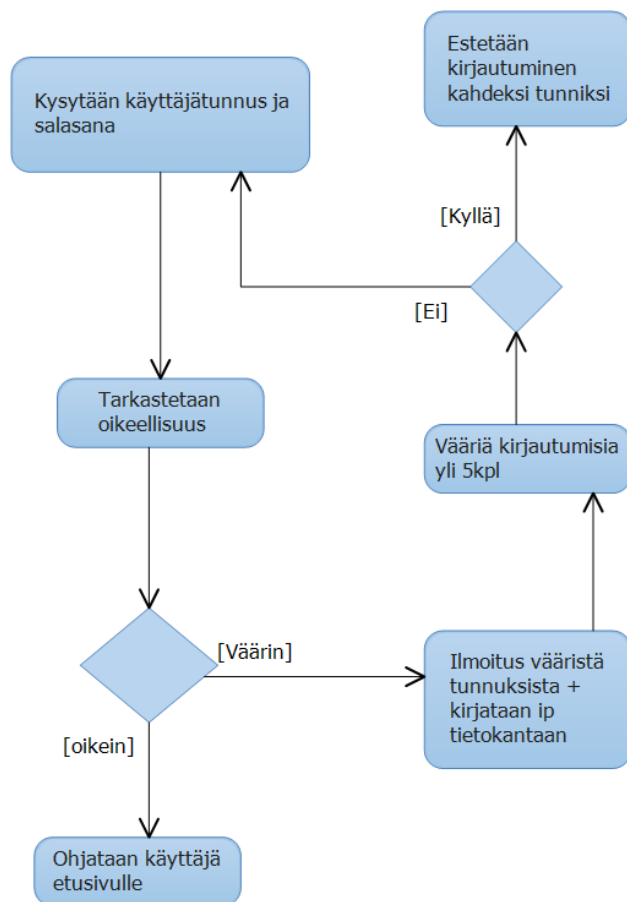
KUVIO 3. Huomion lisääminen on ainut täysin yhteinen toiminto

Käyttöliittymässä erilaisia näkymiä on kymmenen kappaletta sekä laitekorttien näkymiä on 11 kappaletta (liite 1). Näkymiin haettavat tiedot noudetaan tietokannasta sisään kirjautuneen käyttäjätunnuksen perusteella.

## 4.2 Sisäänkirjautuminen

Järjestelmän käyttäminen aloitetaan kirjautumalla järjestelmään sisälle. Käyttäjältä kysytään käyttäjätunnusta sekä salasanaa, joita verrataan tietokannassa oleviin tietoihin. Salasana on kryptattu SHA-512-kryptauksella. Syötetty salasana kryptataan samalla SHA-512-kryptauksella, jonka jälkeen tarkistetaan, löytyykö tietokannasta vastaavaa käyttäjätunnus – salasana yhdistelmää. Jos löytyy, ohjataan käyttäjä pääsivulle, muulloin näytetään virheilmoitus epäonnistuneesta sisäänkirjautumisesta sekä tallennetaan tietokantaan käyttäjän IP-osoite sekä käyttäjätunnus (kuvio 4).

Jos samasta IP-osoitteesta, tai samalla käyttäjätunnuksella, on yritetty epäonnistuneesti kirjautua sisään liian monta kertaa, estetään sisäänkirjautuminen sekä kerrotaan käyttäjälle, että hän on yrittänyt kirjautua liian monta kertaa lyhyen ajan sisällä.

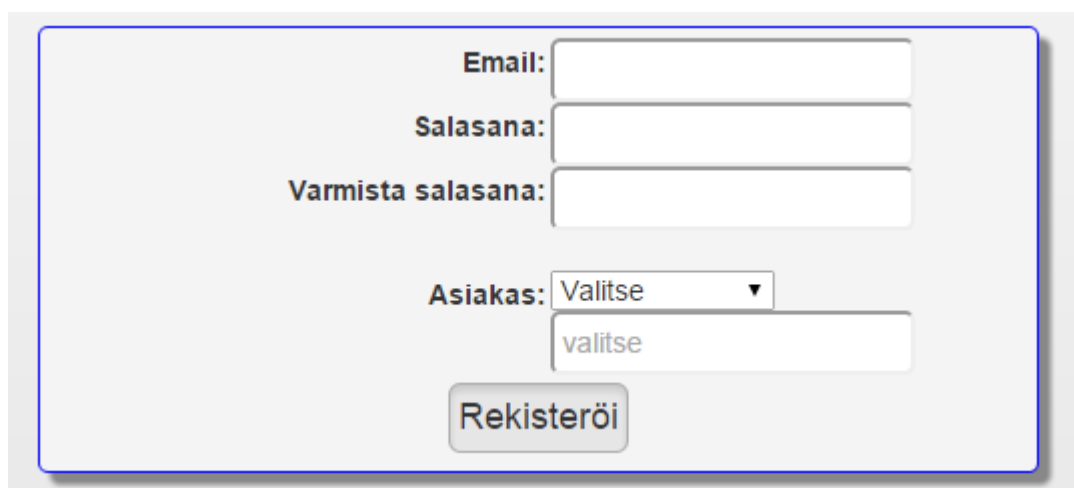


KUVIO 4. Kirjautumisen toimintaperiaate

### 4.3 Uusien käyttäjätunnusten luonti

Vain ylläpidolla on oikeus luoda uusia käyttäjiä järjestelmään. Uuden käyttäjän luonti vaatii järjestelmään sisäänkirjautumisen, jonka jälkeen Hallintapaneelistä voidaan siirtyä sivulle, jossa uusi käyttäjä luodaan.

Uutta käyttäjää lisättäessä annetaan sähköpostiosoite, joka toimii käyttäjätunnuksena, salasana sekä salasanan varmistus ja asiakas, jonka työntekijä käyttäjä on. Käyttäjän oikeudet järjestelmässä riippuvat asiakas-valinnasta, joka käyttäjää luodessa valitaan (kuva 1).



KUVA 1. Käyttäjätunnuksen luomiseen käytettävä lomake

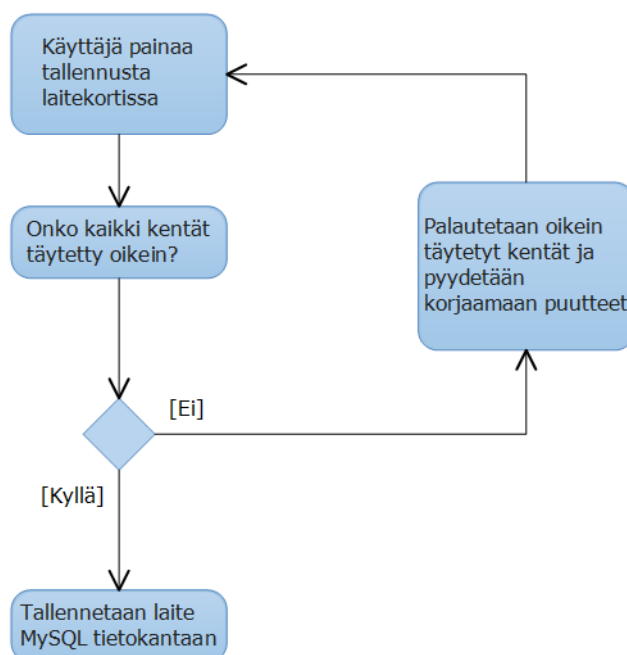
Jokaisella syötteellä on vaatimuksia, mitkä pitää täyttyä, jotta uusi käyttäjä voidaan luoda. Käyttäjätunnuksen, eli salasanan, pitää olla muodoltaan sähköpostiosoite sekä salasanan pitää muodostua vähintään kuudesta merkistä ja sen pitää sisältää vähintään yksi iso kirjain, yksi pieni kirjain sekä yksi numero. Asiakas-valintaa ei voi myöskään jättää tyhjäksi.

### 4.4 Laitteiden lisääminen

Uusien laitteiden lisääminen järjestelmään tapahtuu Laiterekisteri-sivun kautta valitsemalla haluttu laite pudotusvalikosta, joka on näkyvissä vain korkeimman tason käyttäjil-

le. Kun lisättävä laite on valittu, siirrytään sivulle, jossa laitteelle syötetään kaikki tarvittavat ennalta määritetyt tiedot. Jokaiseen kenttään on kirjoitettava vähintään ”-”, tai valittava jokin vaihtoehto kentästä riippuen, jotta laite voidaan lisätä tietokantaan (liite 2).

Kun kentät on täytetty ja käyttäjä painaa ”Tallenna”, tarkistetaan kaikki käyttäjän syötteet virheiden tai tyhjien kenttien varalta. Mikäli kentät on täytetty oikein, kerrotaan käyttäjälle laitteen onnistuneesta lisäämisestä ja ohjataan takaisin Laiterekisteri-sivulle. Muussa tapauksessa käyttäjälle ilmoitetaan virheestä, esimerkiksi tyhjästä kentästä, ja pyydetään tarkistamaan syötekentät (kuvio 3).



KUVIO 3. Laittekortin ollessa puutteellinen palautetaan oikein täytetyt kentät käyttäjälle

#### 4.5 Laitteiden tarkastelu

Järjestelmässä olevia laitteita voi tarkastella Laiterekisteri-sivulla. Sivulla näytetään jokaisen laitteen oleelliset tiedot, kuten laitteen tunnus, nimi, tyyppi, minkä asiakkaan laite se on, elinkaaren tilanne sekä laitteen tulevia tai menneitä huoltoja. Nämä tiedot haetaan tietokannasta kirjautuneen käyttäjän mukaan. Korkeimman tason käyttäjille haetaan tietokannasta kaikki laitteet, muille käyttäjille vain käyttäjän omat laitteet.

Näytettäviä laitteita voi rajata paikkakunnan, alueen tai ”muu tieto”-kentän mukaan, lisäksi korkeimman tason käyttäjät voivat rajata laitteita asiakkaan mukaan. Lisäksi laitelistaa voi järjestää listassa olevien sarakkeiden mukaan, esimerkiksi laitteen tunnuksen mukaan aakkosjärjestykseen.

Jokaisella listattavalla laitteella on myös linkki kyseisen laitteen laitekortille, jossa voi tarkastella laitteen tarkempia tietoja, sekä linkit laitteen huoltojen huoltoraportteihin, mikäli niitä on.

#### **4.6 Laitekortti**

Laitekortilla näytetään yksittäisen laitteen oleelliset tiedot, mahdolliset huomiot mitä laitteelle on lisätty, huollot sekä yleiset tiedostot, esimerkiksi kytkentäkaaviot.

Laitekortilta on myös mahdollista tarkastella laitteen kaikkia tietoja, sekä muokata niitä, jos käyttäjä on korkeimman tason käyttäjä. Tarkastelu ja muokkaaminen tapahtuvat laitekortilla näkyvää ”Muokkaa” tai ”Näytä” -nappia painamalla. Näytettävän napin teksti riippuu käyttäjän tasosta. Napin painallus ohjaa käyttäjän laitteen lisäys näkymään, mutta kaikkiin kenttiin haetaan valmiiksi tiedot tietokannasta. Korkeimman tason käyttäjän on mahdollista päivittää kaikkia laitteen tietoja, jolloin kyseisen laitteen tiedot päivittyvät tietokannassa. Muut käyttäjät voivat vain tarkastella näytettäviä tietoja, mutta eivät voi muokata niitä.

Laitekortin Tiedostot-osiossa näytetään laitteelle lisätyt yleiset tiedostot ja dokumentit, jotka haetaan palvelimelta. Korkeimman tason käyttäjät voivat lisäksi lisätä laitteelle tiedostoja, jotka tallennetaan palvelimelle ja näytetään laitekortilla.

Huollot-osiossa näytetään kerralla kolmen vuoden huollot, niiden päivämäärät sekä tilanteen missä huolto on, eli onko se tehty, tulossa, myöhässä tai vaatiiko se jatkotoimenpiteitä. Lisäksi jokaisella huollolla on linkki huollon huoltoraporttiin, mikäli raportti on lisätty huollolle. Huollot-osiossa on myös nuolinapit, joiden avulla näytettäviä vuosia voi kelata eteen- tai taaksepäin, sekä linkki laitteen huoltokortille.

## 4.7 Huoltokortti

Laitteen huoltokortilla näytetään tarkasteltavan laitteen laitetyyppi sekä tunnus ja listataan laitteelle määritetyt huollot kolme vuotta kerrallaan. Näytettäviä vuosia voi myös kelata eteen- ja taaksepäin nuolinapeista samaan tapaan kuin laitekortillakin. Korkeimman tason käyttäjät voivat lisätä laitteelle uusia huoltoja antamalla lisättävän huollon päivämäärän muodossa dd.mm.yyyy sekä valitsemalla huollon tyyppin pudotusvalikosta (liite 3). Korkeimman tason käyttäjät voivat myös muokata tai poistaa aiemmin lisättyjä huoltoja painamalla huollon nimeä huoltolistauksessa, jolloin huollon tiedot näytetään päivämäärä ja huollon tyyppi -kentissä, ja ”Lisää” -napin tilalla näytetään ”Muokkaa” ja ”Poista” napit.

Jokaisella huollolla, jolle on lisätty huoltoraportti, näytetään myös ”Katso huoltoraportti” nappi, jota painamalla käyttäjä voi ladata huoltoraportin tarkasteltavaksi. Korkeimman tason käyttäjille näytetään ”Lisää huoltoraportti” -nappi, jos huollolle ei vielä ole lisätty huoltoraporttia. Tätä nappia painamalla käyttäjä voi lisätä huollolle huoltoraportin, joka tallennetaan palvelimelle (liite 4).

Huoltokortin lopussa on vielä Tiedostot -osio, jossa kaikki laitteelle lisätyt huoltoraportit on listattu, lisäyspäivämäärän mukaisessa järjestyksessä, uusin ylimmäisenä. Korkeimman tason käyttäjille on näkyvissä myös nappi, josta halutun huoltoraportin voi poistaa palvelimelta.

## 4.8 Huomiot

Jokaisella laitteella on oma Huomiot-osio, johon käyttäjät voivat lisätä huomioita laitetta koskien. Lisätyt huomiot näytetään laitekortilla sille varatussa tekstikentässä, jossa ilmenee huomion lisäyspäivämäärä ja kellonaika, lisääjän käyttäjätunnus sekä itse lisätty huomio.

Jokaisesta lisäystä huomiosta lähetetään myös sähköpostia henkilöille, jotka haluavat ilmoituksen lisätyistä huomioista. Sähköpostista käy ilmi mille laitteelle huomio on lisätty, kuka lisääjä on sekä lisätyn huomion sisältö.

## 4.9 Huoltojen haku

Haku -välilehdellä on mahdollista hakea tietokannassa olevia huoltoja eri hakuparametrien mukaan (kuva 2). Käyttäjä valitsee hakuehdoiksi haluamansa asetukset, jonka jälkeen haetaan tietokannasta kaikki tiedot, jotka täyttävät hakuehdot. Listauksessa näytetään laitteen perustiedot, jolle huolto on merkitty, sekä huollon tiedot.

Listauksessa on myös linkit jokaisen haetun laitteen laitekortille ja huoltokortille. Listauksista voi järjestää uudelleen minkä tahansa sarakkeen, esimerkiksi huoltojen päivämäärän tai laitteen tunnuksen mukaan.

Laite			
Tyyppi	Tunnus	Laite	
<input type="text"/>	<input type="text"/>	Kaikki	
Asiakas		Muu tieto	
Asiakas	Paikkakunta	Alue	Muu tieto
Kaikki	Kaikki	Kaikki	Kaikki
Tila		Elinkaari	
Tila	Huollon status	Huollon nimi	Elinkaari
Kaikki	Kaikki	Kaikki	Kaikki
Ajankohta			
Aloituspvm	Lopetuspvm		
1.1.1970	1.1.2100	haku	

KUVA 2. Huoltojen hakuun käytettävät kentät

## 4.10 Dokumentit

Dokumentit -välilehdeltä löytyvät järjestelmään lisätyt yleiset dokumentit, jotka eivät liity suoraan mihinkään laitteeseen tai huoltoon, esimerkiksi hinnastot. Jokaisella asiakasryhmällä on omat dokumentit, joita ei näytetä muille asiakkaille. Korkeimman tason käyttäjät voivat kuitenkin tarkastella minkä tahansa asiakasryhmän dokumentteja valitsemalla haluttu asiakasryhmä pudotusvalikosta.

Dokumenttien listauksessa dokumentin nimeä painamalla voi dokumentin ladata palvelimelta tarkastelua varten.



## **4.11 Yhteystiedot**

Jokaiselle asiakkaalle määritellään omat yhteyshenkilöt, jotka näytetään Yhteystiedot -välilehdellä. Näkyviä tietoja ovat muun muassa titteli, paikkakunta, sähköpostiosoite sekä puhelinnumero.

Sivulla näytetään aina asiakkaan yhteyshenkilöiksi määritellyt yhteystiedot vasemmalla, sekä sisään kirjautuneen käyttäjän asiakasyrityksen yhteystiedot oikealla. Korkeimman tason käyttäjät voivat pudotusvalikolla valita, minkä asiakasyrityksen yhteystietoja oikeassa sarakkeessa näytetään. Muut käyttäjät näkevät vain heille tarkoitettut yhteystiedot.

## **4.12 Hallintapaneeli**

### **4.12.1 Käyttäjien lisäys ja muokkaus**

Korkeimman tason käyttäjillä on mahdollisuus muokata järjestelmään rekisteröityjä käyttäjiä hallintapaneelin kautta. Käyttäjää pyydetään ensin hakemaan haluttua rekisteröityä käyttäjää sähköpostin tai asiakkaan mukaan tai listaamaan kaikki järjestelmän käyttäjät. Kun haluttu käyttäjä on valittu, voidaan sen tietoja muokata.

Tietoja muokattaessa pätee samat säännöt kuin uutta käyttäjää lisätessä eli salasanan pitää sisältää vähintään kuusi merkkiä ja sisältää ainakin yksi iso kirjain, yksi pieni kirjain sekä yksi numero, sanoin käyttäjätunnuksen pitää olla sähköpostiosoite.

Mikäli käyttäjä muokkaa omia tietojaan, ohjataan käyttäjä muutosten jälkeen takaisin sisäänkirjautumissivulle, jotta tehdyt muutokset astuvat heti voimaan.

#### **4.12.2 Dokumenttien lisäys ja muokkaus**

Hallintapaneelin kautta voidaan halutulle asiakkaalle lisätä yleisiä dokumentteja sekä muokata olemassa olevia dokumentteja. Vain korkeimman tason käyttäjä voi suorittaa nämä toiminnot.

Mikäli käyttäjä on lisäämässä uutta dokumenttia, valitsee hän ensin asiakkaan, kenelle dokumenttia ollaan lisäämässä ja sen jälkeen tiedoston, jonka haluaa lisätä. Lisäyksen jälkeen dokumentti näkyy valituille asiakkaille Dokumentit -välilehdellä.

Olemassa olevien dokumenttien muokkaus tapahtuu valitsemalla ensin asiakas, jonka dokumentteja halutaan muokata. Valinnan jälkeen käyttäjälle listataan kaikki valitun asiakkaan dokumentit, sekä toimenpiteet, mitä dokumentille voi tehdä, kuten dokumentin poistaminen järjestelmästä.

#### **4.12.3 Yhteystietojen lisäys ja muokkaus**

Järjestelmään lisätään uusia, sekä muokataan olemassa olevia yhteystietoja hallintapaneelin kautta. Yhteystietoa lisätessä yhteystiedolle pitää antaa nimi, paikkakunta, titteli, sähköposti sekä asiakas, jonka yhteyshenkilö lisättävä henkilö on. Lisäksi on mahdollista antaa korkeintaan neljä tekniikka-aluetta, joista lisättävä henkilö vastaa.

Mikäli yhteyshenkilön asiakas-kenttään valitsee järjestelmän käyttäjän, pyydetään lisääjää valitsemaan myös minkä asiakkaiden yhteyshenkilö lisättävä henkilö on. Valintoja voi tehdä useita. Tällöin lisättävä henkilö näytetään Yhteystiedot -sivulla niille käyttäjille, jotka kuuluvat valittuihin asiakasryhmiin.

Yhteystietojen muokkaus tapahtuu samalla tavalla kuin lisäyskin, mutta jokainen syötökenttä täytetään valmiiksi tietokannasta haettavilla tiedoilla, joita voi sen jälkeen muokata. Lisäksi on mahdollista lisätä yhteystiedolle kuva, joka näytetään Yhteystiedot -sivulla muiden tietojen vieressä.

#### 4.13 Huoltotriggeri

Järjestelmä lähettää joka maanantaiaamu kello seitsemän automaattisesti sähköpostiviestin tulevista huolloista kaikille niille yhteyshenkilöille, jotka ovat määritelty saamaan sähköpostia tulevista huolloista.

Korkeimman tason käyttäjät voivat muokata sähköpostiviestien saajia hallintapaneelistä Sähköpostilista -kohdasta. Käyttäjä valitsee ensin yhteystiedon, jonka sähköpostituslistaa halutaan muokata. Tämän jälkeen tietokannasta haetaan valitun yhteystiedon sähköpostituslista, joka näytetään käyttäjälle (kuva 3).

Yhteystiedolle valitaan asiakas, sekä paikkakunta, minkä laitteista lähetetään sähköpostia. Samalle yhteystiedolle on mahdollista valita useita asiakas-paikkakunta yhdistelmiä, jonka seurauksena valittu yhteyshenkilö saa sähköpostia kaikista valituiden asiakkaiden laitteiden huolloista.

Sähköpostilista -sivulla määritellään samalla myös se, että lähetetäänkö henkilölle sähköpostia laitteille lisätyistä huomioista.

Valitse yhteystieto, jonka sähköpostilistaa haluat muokata:

Ylläpito Ylläpito ▼

Valitse asiakkaat

Valitse asiakkaat		Lähetä postia tulevista huolloista	Lähetä postia uusista huomioista
Asiakas 2, Tampere	Asiakas 2, Tampere ▼	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Asiakas 2, Helsinki	Asiakas 2, Helsinki ▼	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Valitse	Valitse ▼	<input type="checkbox"/>	<input type="checkbox"/>

Tallenna

KUVA 3. Sähköpostilistan ulkoasu

## 5 KESKEISET OSAT

### 5.1 Tiedostojen tallennus

Järjestelmään talletetaan monia erityyppisiä tiedostoja, joiden jokaisen talletus hoidetaan eri tavalla riippuen siitä, minkälaista tiedostoa ollaan tallentamassa. Varsinaiselle tiedostotyyppille, esimerkiksi .pdf tai .txt, ei ole asetettu rajoituksia asiakkaan pyynnöstä. Mahdollisia tiedostotyyppejä on monenlaisia, sekä ne saattavat muuttua useinkin, ettei rajoittaminen ole ylläpidollisista syistä kannattavaa.

Mikäli jollekin laitteelle ollaan lisäämässä tiedostoa, joko huoltoraporttia tai yleistä laitteeseen liittyvää tiedostoa, tapahtuu tiedoston lisäys palvelimelle finput.php:n avulla. Eri laitteiden tiedostot talletetaan palvelimelle omiin kansioihinsa, eroteltuna tietokannasta tulevalle uniikilla laitteen id:llä. Lisäksi jos lisättävä tiedosto on huoltoraportti, lisätään ne laitteen kansion sisälle luotavaan ”huoltoraportit”-kansioon.

Palvelimen yleiset dokumentit ja tiedostot, jotka eivät liity suoranaisesti mihinkään laitteeseen, lisätään dinput.php:n avulla (kuva 4). Eri asiakkaille tarkoitetut tiedostot erotellaan omiin kansioihinsa asiakkaan nimen mukaan.

```

if(isset($_POST["asiakas_valinta"])) {
    $asiakas = $_POST["asiakas_valinta"];
    $uploadOk = 1;
    $target_dir = "documents/" . $asiakas;
    if (!file_exists($target_dir)) {
        mkdir($target_dir, 0777, true);
    }
    $target_file = $target_dir . basename($_FILES["uploadedfile"]["name"]);
    $fileType = pathinfo($target_file,PATHINFO_EXTENSION);
    if (file_exists($target_file)) {
        $uploadOk = 0;
    }
    if ($uploadOk == 0) {
        return false;
    }
    else{
        if (move_uploaded_file($_FILES["uploadedfile"]["tmp_name"], $target_file)) {
            return true;
        }
        else {
            return false;
        }
    }
}
else{
    return false;
}

exit;

```

KUVA 4. Esimerkki dokumenttien lisäämisestä palvelimelle

Lisätyille yhteystiedoille on mahdollista lisätä kuva, joka näytetään Yhteystiedot - välilehdellä. Kuvien lisäykseen on tehty tarkistus, joka vaatii, että lisättävä tiedosto on tiedostotyyppiltään kuva, kuten .jpg tai .png. Eri yhteystietojen kuvat erotellaan palvelimella omiin kansioihinsa, tietokannasta tulevan yhteystiedon uniikin id:n avulla. Lisäksi kuva tallennetaan nimellä ”yhteystiedon\_nimi.kuvan tiedostotyyppi”, esimerkiksi ”Mikko\_Mallikas.png”.

## 5.2 Laitteiden lisääminen

Laitteiden lisäys tietokantaan suoritetaan input.php:n avulla. Siitä pyrittiin tekemään mahdollisimman dynaaminen jatkokehitystä ja ylläpitoa silmällä pitäen. Vaikka jollekin laitetyypille lisättäisiin uusi kenttä, jota ei aikaisemmin ole ollut, ei input.php:tä tarvitse muokata.

Laitetta lisätessä tarkistetaan ensimmäisenä onko kaikki vaadittavat kentät täytetty, vähintään viivalla. Jos tyhjiä kenttiä löytyy, ei laitteen lisäämistä toteuteta, vaan pyydetään käyttäjää täyttämään kaikki vaaditut kentät. Seuraavaksi käydään käyttäjän täyttämät syöttökentät läpi ja asetetaan ne taulukkoon jatkokäyttöä varten. Taulukko on kaksiulotteinen taulukko, johon tallennetaan vaadittujen tietokannan taulujen nimet, taulun kenttien nimet sekä kenttien sisältö.

Taulukko järjestetään siten, että tiedot saadaan lisättyä tietokantaan oikeassa järjestyksessä, samalla kun taulua käydään läpi. Taulujen liitosten takia tämä on välttämätöntä, esimerkiksi laitetyypille ominaiset tiedot pitää tallentaa tietokantaan ennen laitteen yleisten tietojen lisäämistä.

Mikäli laitteen tietojen lisäys epäonnistuu jossakin vaiheessa, lopetetaan sen lisäys ja perutaan mahdolliset aiemmat tallennukset, mitä laitetta lisätessä on tehty. Tällä estetään virheellisten ja satunnaisten tietojen päätyminen tietokantaan.

### 5.3 Hallintapaneeli

Hallintapaneelin toiminta on estetty lähes kokonaan kaikilta muilta, paitsi korkeimman tason käyttäjiltä. Käyttäjän taso tarkistetaan, eikä kaikkia hallintapaneelin toimintoja näytetä, mikäli käyttäjä ei omaa korkeimman tason tunnuksia. Varmuuden vuoksi myös jokaista toimintoa tehdessä tarkistetaan käyttäjän taso siltä varalta, että käyttäjä pääsisikin käsiksi toimintoihin, joihin käyttäjän ei pitäisi päästä.

Hallintapaneeli.php:n kautta käyttäjä valitsee haluamansa toiminnon, joka suoritetaan hallintapaneeli\_views.php:ssä. Mikäli toiminto on monivaiheinen, siirrytään hallintapaneeli\_views.php:stä eteenpäin, esimerkiksi muokkaa\_kayttaja.php:n, missä halutun käyttäjän tietoja voi muokata. Projektin alussa kaikki hallintapaneelin toiminnot oli tarkoitus suorittaa suoraan hallintapaneeli\_views.php:n kautta, mutta projektin edetessä huomattiin, että on parempi suorittaa eri toiminnot eri tiedostoissa. Tämä helpottaa ylläpidollista toimintaa, kun yhden tiedoston pituus ja sen suorittamat toiminnot eivät paisu hirveän suuriksi.

### 5.4 Sähköpostilista

Sahkopostilista.php ohjaa tietokannassa olevaa taulua, joka määrittelee kenelle järjestelmä lähettää sähköposteja määrättyinä hetkinä. Tietokannan tauluun tallennetaan yhteystieto – asiakas -yhdistelmiä, joita weeklymailsendscript.php käy läpi.

Palvelin suorittaa omatoimisesti weeklymailsendscript.php -scriptin, joka etsii tietokannasta ne laitteet ja huollot, jotka ovat tulossa kahden viikon päästä alkavalla viikolla. Tämän jälkeen haetaan sähköpostilistasta ne yhteystiedot, jotka ovat määritelty saamaan sähköpostia kyseisten laitteiden huolloista ja lähetetään heille sähköposti. Sähköpostiviestistä ilmenee, mille laitteelle huolto on tulossa, mikä huolto se on sekä huollon tarkka päivämäärä.

Sähköpostilistan kautta hoidetaan myös sähköpostiviestien lähetys kun laitteille lisätään huomioita. Aina, kun laitekortilla lisätään laitteelle huomio, haetaan sähköpostilistasta ne henkilöt, jotka haluavat viestiä kyseiselle laitteelle lisätyistä huomioista. Tarkistus tapahtuu laitteen omistajan sekä sijainnin mukaan.

Varsinainen sähköpostiviestien lähetys tehdään PHP:n omalla mail() -funktioilla (php.net/manual/en/function.mail.php, 2015). Tähän ratkaisuun päädyttiin siksi, ettei se vaadi ylimääräisiä kirjastoja, ja sen asennus on helppoa. Ainoa vaadittava toimenpide on sendmail-paketin asennus palvelimelle, jonka jälkeen jo pelkät oletusasetukset riittävät sähköpostiviestien lähettämiseen.

## 5.5 .htaccess

.htaccess:n avulla sivustolla estetään lisättyjen tiedostojen ja raporttien suora aukaiseminen osoiteriviltä. Vain tiedostoja, joiden tyyppi on .php tai .html, voidaan avata suoraan internetselaimen osoiteriviltä. Apachessa on myös oletuksena päällä hakemistojen indeksointi, joka estetään .htaccess tiedostolla.(kuva 5)

```
6 DirectoryIndex index.php
7 RewriteEngine on
8 Options All -Indexes
9 RewriteCond %{HTTP_REFERER} !^http(s)?://(www\.)?domain.fi [NC]
10 RewriteRule !\.(html|php)$ - [NC,F,L]
```

KUVA 5. Koodiesimerkki .htaccess –tiedostosta

Lisäksi .htaccess:n avulla pakotetaan kaikkien palvelimelle tallennettujen tiedostojen, kuten raporttien, lataus. Html:ssä on tiedostojen linkeille olemassa ”download” parametri, mutta kaikki selaimet, joilla järjestelmän on toimittava, ei tue sitä. Tämän takia latauksen pakotus on tehty .htaccess:lla. Järjestelmään tallennetaan useita eri tiedostomuodossa olevia tiedostoja, joten kaikkia tiedostomuotoja, joita jatkossa tullaan käyttämään, on mahdotonta listata. .htacceess tiedostoon on listattu ne tiedostomuodot, jota ei ladata järjestelmästä(kuva 6).

```
1
2 <FilesMatch "(?!\. (gif|jpe?g|png|js|css|swf|php|ico|txt|xml|html?))$" >
3   Header set Content-Disposition attachment
4 </FilesMatch>
5
```

KUVA 6. Rivillä 2 esitetään tiedostot, joihin pakotusta ei ole asetettu

## 5.6 Tietokanta

Järjestelmän taustalla olevaa tietokantaa suunniteltiin uudestaan ja muokattiin muutama otteeseen projektin edetessä. Sen suunnittelua vaikeutti tarkan tiedon puute siitä mitä kaikkea sinne halutaan tai pitää tallentaa.

Tietokannan ensimmäinen versio sisälsi yli 50 taulua, ja siitä luovuttiin hyvin aikaisessa vaiheessa. Sen tilalle suunniteltiin uusi, noin 10 taulua sisältävä kanta, joka on pohjana lopullisen järjestelmän tietokannalle. Projektin edetessä tietokantaan lisättiin vielä muutama yksittäinen taulu, sekä tehtiin joitakin lisäyksiä jo olemassa oleviin tauluihin. Lopputuloksena tietokanta sisältää 20 taulua.

Lähes koko järjestelmä pyörii laitteiden yleisiä tietoja sisältävän taulun ympärillä. Siihen yhdistyy eri laitetyyppien omat taulut, jotka sisältävät eri laitetyyppien uniikkeja tietoja. Yhdessä näillä tauluilla saadaan tallennettua kaikkien laitteiden tiedot tietokantaan ilman, että tietokannan tauluihin tulisi suuri määrä tyhjiä kenttiä.

Tietokannan keskiössä olevaan laitetauluun yhdistyy myös kaksi taulua, joihin tallennetaan eri laitteille lisätyt huomiot sekä laitteelle tehtävät huollot. Näin saadaan yksilöityä huollot ja huomiot aina yksittäiselle laitteelle ja niiden hakeminen sekä näyttäminen onnistuvat helposti.

Lisäksi tietokannassa on oma taulunsa yhteystietojen tallentamiseen, joita järjestelmän verkkosivuilla näytetään. Siihen liittyvällä taululla tallennetaan lista asiakkaista, joiden laitteista yhteystiedolle tulee lähettää sähköpostia.

Tietokannassa on myös omat irralliset taulunsa rekisteröityneiden käyttäjien tallentamiseen sekä epäonnistuneiden sisäänkirjautumisyritysten tallentamiseen. Näiden avulla estetään satunnaisten ihmisten tai hakurobottien pääsy järjestelmään.

Tietokantaan ei tallenneta mitään arkaluontoista tietoa, kuten henkilö- tai luottokorttitietoja.



## 5.7 Tablesorter

Tablesorteria käytetään laiteluettelon ja huoltohaun tuloksien järjestyyn. Tablesorter osaa oletusasetuksilla järjestää pystysarakkeita numero- tai aakkosjärjestykseen, mutta osa kentistä on sellaisia, joihin tarvitsee muokata hieman lisäosan toimintaa.

Suomalaiseen tapaan päivämääräjärjestys on saatava muotoon pp.kk.vvvv, mutta tässä muodossa olevat päivämäärät järjestettäisiin käyttäen päiviä järjestyslukuna, joka johtaa pystysarakkeiden epäloogiseen järjestykseen. Tätä ongelmaa tutkiessa todettiin, että samaan ongelmaan on törmätty muissakin projekteissa internetin keskustelupalstoilla. Tämän takia tehtiin tablesorteriin jQueryllä jäsentäjä, joka järjestää sarakkeen niin, että vuodet ja kuukaudet priorisoidaan ennen päivämääriä (kuva 7). Tätä jäsentäjää käytetään myös laiteluettelossa, jossa ainut ero huoltohaun jäsentäjään on pystysarakkeen järjestysnumero.

```

3  = $.tablesorter.addParser({
4      id: 'dmy',
5      is: function(s) {
6          return false;
7      },
8      format: function(s) {
9          var date = s.split('.');
10         return new Date(date[2], date[1], date[0]).getTime();
11     },
12     type: 'numeric'
13 });
14
15 = $(function() {
16     $('#kortti').tablesorter({
17
18         headers: {
19             9: { sorter: "dmy" }
20         }
21     });
22 });
23
24
25

```

KUVA 7. Hakutoiminnossa käytetty jäsentäjä

Laiteluettelossa esitetään myös laitteen elinkaaren tila taulukossa, jota pitää pystyä järjestelemään. Tätä varten asetettiin jokaiseen elinkaaren lohkokoon numerot yhdestä neljään, jotka vastaavat elinkaaren tilaa. Numero on samalla värillä kuin lohkon tausta, joka auttaa myös elinkaaren näkymiseen, jos listasta otetaan mustavalkotuloste.

## **6 JATKOTOIMENPITEET**

### **6.1 Ylläpito**

Huoltoseurantajärjestelmä on Tietotekniikan Tuotekehitys ja Innovointi Verstaan palvelimella Tampereen Ammattikorkeakoulun tiloissa. Varsinaisen ylläpidon hoitaa TKI-Verstas.

Järjestelmän ylläpidosta on tämän dokumentin kirjoitushetkellä käynnissä neuvottelut järjestelmän tilaajan ja TKI-Verstaan välillä yksityiskohtien hiomiseksi kaikkia osapuolia tyydyttävään muotoon.

### **6.2 Asiakkaan käytössä ilmenevät asiat**

Järjestelmän käyttöönoton jälkeen ilmenevät ongelmat korjataan asiakkaan toiveiden mukaisesti ylläpitoon rekrytoidun Tampereen Ammattikorkeakoulun opiskelijan toimesta. Mahdollisesta jatkokehityksestä sovitaan erikseen.

Vikakorjaukset pyritään tekemään mahdollisimman nopeasti. Kuitenkin mikäli korjauspyyntö sisältää uusien ominaisuuksien lisäämisiä, sovitaan sen toteuttamisesta erikseen.

### **6.3 Uusien ominaisuuksien kehitys**

Järjestelmä pyrittiin toteuttamaan niin, että sen jatkokehitys olisi helppoa, eikä vaadi jatkokehittäjältä laajaa ohjelmointikokemusta. Mahdollisia jatkokehitys mahdollisuuksia on lähes loputon määrä. Jo ennen järjestelmän varsinaista käyttöönottoa asiakas esitti muutamia toiveita, mitä järjestelmään voisi jatkokehityksen puitteissa lisätä.

Tietokannassa olevan datan perusteella pystyisi generoimaan diagrammeja tai muita esityksiä asiakkaan laitteista tai huoltojen suorituksista. Tämä ominaisuus on herättänyt mielenkiintoa asiakkaiden keskuudessa. Ominaisuus on täysin toteutettavissa esimerkiksi Raphaël nimisen JavaScript kirjaston avulla ([raphaeljs.com](http://raphaeljs.com), 2015).

Myös halutun asiakkaan laiterekisterin kopiointi tai siirto Excel-taulukkoon on toivottu ominaisuus. Mikäli asiakas haluaa lopettaa yhteistyön järjestelmää käyttävän tahon kanssa, pitää asiakkaalla olla keino saada järjestelmään tallennetut tiedot ulos. Tätä dokumenttia kirjoittaessa tämä ei ole suoraan mahdollista, mutta tarpeen vaatiessa halutut tiedot saadaan järjestelmästä haettua ylläpidon kautta.

## 7 YHTEENVETO

Projektin aikana luotu huoltoseurantajärjestelmän ensimmäinen versio täytti asiakkaan asettamat vaatimukset ja toiveet. Sen avulla asiakas pystyy korvaamaan vanhan Excel- taulukko pohjaisen menetelmän huoltojen ja laitteiden seurannassa. Työn edetessä pidettiin aktiivisesti yhteyttä asiakkaaseen ja kuunneltiin heidän ideoita sekä ehdotuksia ja sen jälkeen tehtiin tarvittavia muutoksia. Tästä toimintatavasta tuli asiakkaalta hyvää palautetta. Myös järjestelmän käyttöliittymä onnistuttiin pitämään yksinkertaisena ja intuitiivisena.

Yksi isoista haasteista ollut ylläpidettävyyys onnistuttiin toteuttamaan pitäytymällä tunnetuissa ohjelmointikielissä, joihin löytyy paljon esimerkkejä ja manuaaleja muun muassa internetin keskustelupalstoilta. Ohjelmakoodi itsessään pyrittiin pitämään mahdollisimman selkeänä ja helppolukuisena, jotta ylläpitäjän ei tarvitse omata laajaa ohjelmointikokemusta, vaan pelkät perustiedot riittävät.

Toteutuksessa käytetyt ohjelmointikielet ja ohjelmointityyli rupeavat olemaan vanhanaikaisia ja kohtuullisen kankeita. Eri näkymät ja sisällön näyttäminen tapahtuvan hyvin suurelta osin HTML formien post -ominaisuuden avulla, jolloin koko sivu joudutaan lataamaan palvelimelta uudestaan. Nykyaikaisempi menetelmä olisi käyttää esimerkiksi Angular.js:ää, jonka avulla ladattaisiin uudestaan vain se osa sivua, mikä muuttuu. Kuitenkin ylläpitoa silmällä pitäen päädyttiin käyttämään ohjelmointikieliä, joiden perusteet opetetaan hyvinkin aikaisessa vaiheessa tietotekniikan opinnoissa.

Järjestelmän jatkokehitysmahdollisuuksia on lähes rajattomasti ja asiakkaalla on mielessä joitakin kehitysehdotuksia, jotka luultavasti toteutetaan huoltoseurantajärjestelmän seuraaviin versioihin. Erilaisten piirakoiden piirtäminen käyttäjälle tietokannassa olevan tiedon perusteella sekä tietojen saaminen tietokannasta vanhan malliseen Excel – taulukkoon ovat luultavasti ensimmäisiä asioita, mitä asiakas haluaa järjestelmään lisätä.

Tehty järjestelmä räätälöitiin täysin asiakkaan käytön mukaiseksi, eikä sitä ole tarkoitus lähteä tarjoamaan muille asiakkaille. Räätälöinti johtui asiakkaan toiveesta saada jotain sellaista, mitä muilla alalla toimivilla yrityksillä ei vielä ole ja jota ei olisi helppo suo-

raan kopioida. Ilman tätä asiakkaan toivetta järjestelmästä olisi ollut mahdollista kehittää yleispätevä versio kenen tahansa käyttöön.

Tätä dokumenttia kirjoittaessa sopimusneuvottelut järjestelmän ylläpidosta ja jatkokehityksestä ovat käynnissä asiakkaan ja TKI-Verstaan välillä. Tarkoituksena on, että järjestelmän jatkotoimenpiteistä tulee vastaamaan TAMK:n tietotekniikan opiskelija tai opiskelijat. Koko järjestelmän toteutustavassa pidettiin tätä näkökulmaa mielessä, jotta jatkotoimenpiteistä voisi vastata lähes kuka tahansa tietotekniikkaa opiskeleva opiskelija jolla on perustiedot ohjelmoinnista.

Loppujen lopuksi projekti oli onnistunut ja asetettuihin tavoitteisiin päästiin, vaikka aikataulu venyikin hivenen alkuperäisestä suunnitellusta. Järjestelmän avulla asiakas pystyy myös välttämään suurempien toiminnanohjausjärjestelmien käyttöönoton. Kuitenkin mikäli järjestelmää lähdetäisiin nyt miettimään ja rakentamaan uudestaan, käytettäisiin varsinaiseen suunnitteluun hivenen enemmän aikaa. Lisäksi ohjelmointikielet ja tyyli muuttuisi johonkin nykyaikaisempaan, esimerkiksi client- ja server -puoli Angular.js:llä ja pohjalle NoSQL tietokanta. Tämä kuitenkin vaikeuttaisi ylläpidon toimintaa, elleivät tietotekniikan koulutusohjelman opetusaiheet verkko-ohjelmoinnin osalta vaihdu.

## LÄHTEET

Notepad++ Home. Luettu 13.5.2015.

<https://notepad-plus-plus.org/>

Sublime Text: The text editor you'll fall in love with. Luettu 13.5.2015.

<http://www.sublimetext.com/>

AngularJS – Superheroic JavaScript MVW Framework. Luettu 23.5.2015

<https://angularjs.org/>

MEAN – Full-Stack JavaScript. Luettu 23.5.2015

<http://mean.io/>

Node.js

<https://nodejs.org/>

bootstrap/LICENSE at master · twbs/bootstrap · GitHub. Luettu 13.5.2015.

<https://github.com/twbs/bootstrap/blob/master/LICENSE>

Foundation | The Most Advanced Responsive Front-end Framework from ZURB.  
Luettu 13.5.2015.

<http://foundation.zurb.com/>

Gumby - A Flexible, Responsive CSS Framework - Powered by Sass. Luettu 13.5.2015.

<http://gumbyframework.com/>

jQuery plugin: Tablesorter 2.0. Luettu 15.5.2015.

<http://tablesorter.com/docs/>

MySQL – What is MySQL. Luettu 24.5.2015.

<https://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>

The Apache HTTP Server Project. Luettu 15.5.2015.

<http://httpd.apache.org/>

TortoiseSVN. Luettu 15.5.2015.

<http://tortoisesvn.net/>

Git. Luettu 15.5.2015.

<http://git-scm.com/>

MySQL :: MySQL Workbench. Luettu 15.5.2015.

<http://www.mysql.com/products/workbench/>

phpMyAdmin. Luettu 18.5.2015.

[http://www.phpmyadmin.net/home\\_page/](http://www.phpmyadmin.net/home_page/)

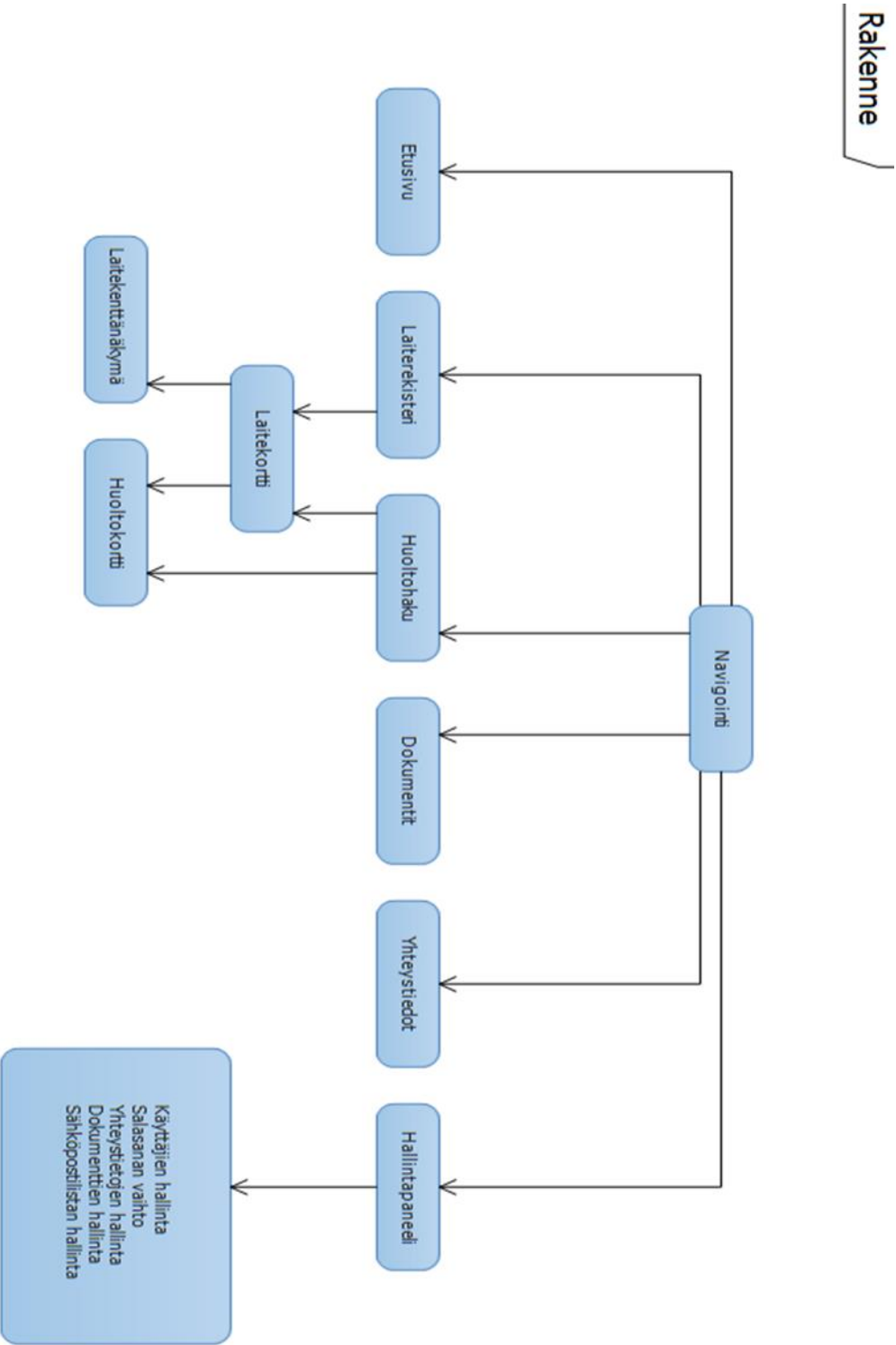
PHP: mail – Manual. Luettu 18.5.2015.

<http://php.net/manual/en/function.mail.php>

Raphaël—JavaScript Library. Luettu 16.5.2015.  
<http://raphaeljs.com/>

LIITTEET

Liite 1. Järjestelmän navigoinnin rakenne





## Liite 2. Käyttötapaus – Laitteen lisäys

**Nimi:** Laitteen lisäys

**Suorittaja:** Korkeimman tason käyttäjä

**Esiehdot:** Käyttäjä omistaa korkeimman tason käyttäjätunnukset.

**Kuvaus:** Käyttäjä avaa huoltoseurantajärjestelmän sisäänkirjautumissivun selaimellaan. Hän kirjoittaa käyttäjänimensä ja salasansansa niihin varatuille kentille ja kirjautuu järjestelmään sisään painamalla ”Login”-nappia.

Selaimeen aukeaa järjestelmän etusivunäkymä. Käyttäjä painaa ”Laiterekisteri”-linkkiä, jolloin aukeaa näkymä, jossa listataan kaikki halutut laitteet. Sivulta löytyy linkki ”Lisää laitteita” jossa pudotusvalikosta valitaan lisättävän laitteen tyyppi. Kun lisättävää laitetyyppiä painaa, aukeaa käyttäjälle näkymä, jossa on kaikki kyseiselle laitetyypille vaaditut tietokentät. Käyttäjä täyttää kaikki näytettävät kentät vähintään viivalla ”-”, valitsee pudotusvalikoista jonkin vaihtoehdon ja tekee vaaditut valinnat radionapeilla. Tämän jälkeen käyttäjä painaa ”Tallenna”-nappia. Käyttäjälle ilmoitetaan että tallennus onnistui ja ohjataan hänet takaisin laiterekisterisivulle.

**Poikkeukset:** Sisäänkirjautuminen epäonnistuu: kirjautumistiedot ovat väärät.

Laitteen lisäys epäonnistuu: kaikkia vaadittuja kenttiä ei täytetty, pudotusvalikoista ei ole tehty valintoja, radionapeista ei ole tehty valintaa.

**Lopputulos:** Käyttäjä on lisännyt järjestelmään uuden laitteen ja se on tallennettu tietokantaan.

### Liite 3. Käyttötapaus – Huollon lisäys

**Nimi:** Huollon lisäys

**Suorittaja:** Korkeimman tason käyttäjä

**Esiehdot:** Käyttäjä omistaa korkeimman tason käyttäjätunnukset, järjestelmään on lisätty laite.

**Kuvaus:** Käyttäjä avaa huoltoseurantajärjestelmän sisäänkirjautumissivun selaimellaan. Hän kirjoittaa käyttäjänimensä ja salasansa niihin varatuille kentille ja kirjautuu järjestelmään sisään painamalla ”Login”-nappia.

Selaimeen aukeaa järjestelmän etusivunäkymä. Käyttäjä painaa ”Laiterekisteri”-linkkiä, jolloin aukeaa näkymä, jossa listataan kaikki halutut laitteet. Tämän jälkeen käyttäjä painaa laiterekisterisivulla ”Tarkastele laitteita”-pudotusvalikosta lisäämäänsä laitetyyppiä, tai vaihtoehtoisesti ”kaikki”. Sivulle haetaan tietokannasta pudotusvalikon valinnan mukaisesti valitun laitetyyppin laitteet ja näytetään niiden perustiedot taulukossa. ”Laitteen nimi”-sarakkeesta käyttäjä painaa laitteen nimeä, jolloin näytetään uudella sivulla tarkempaa tietoa valitusta laitteesta.

”Huollot”-väliotsikkoa painamalla käyttäjä avaa laitteen huoltokortin, jossa hän voi lisätä laitteelle huollon. Huollon nimi valitaan pudotusvalikosta ja sille annetaan päivämäärä muodossa pp.kk.vvvv. Lopuksi painetaan ”Lisää”-nappia, joka lisää huollon tietokantaan sekä näyttää lisätyn huollon huoltokortin huolloille varatussa taulukossa.

**Poikkeukset:** Sisäänkirjautuminen epäonnistuu: kirjautumistiedot ovat väärät.

Huollon lisäys epäonnistuu: Pudotusvalikosta ei ole valittu huollon tyyppiä, huollon päivämäärä on annettu väärässä muodossa tai virheellisesti.

**Lopputulos:** Käyttäjä on lisännyt valitulle laitteelle huollon ja se on tallennettu tietokantaan.

#### Liite 4. Käyttötapaus – Huollon merkkäminen tehdyksi ja huoltoraportin lisääminen

**Nimi:** Huollon merkkäminen tehdyksi ja huoltoraportin lisääminen

**Suorittaja:** Korkeimman tason käyttäjä

**Esiehdot:** Käyttäjä omistaa korkeimman tason käyttäjätunnukset, järjestelmään on lisätty laite, laitteelle on lisätty huolto.

**Kuvaus:** Käyttäjä avaa huoltoseurantajärjestelmän sisäänkirjautumissivun selaimellaan. Hän kirjoittaa käyttäjänimensä ja salasansa niihin varatuille kentille ja kirjautuu järjestelmään sisään painamalla ”Login”-nappia.

Käyttäjä avaa ”Laiterekisteri”-sivun ja valitsee haluamansa laitteen painamalla laitteen nimeä. Aukeavalta sivulta ”Huollot”-väliotsikkoa painamalla käyttäjä avaa laitteen huoltokortin, jossa näytetään laitteelle merkityt huollot, kolme vuotta kerrallaan.

Käyttäjä painaa ”Tehty”- ruutua valitun huollon kohdalla, jolloin järjestelmä kysyy vaaditaanko jatkotoimenpiteitä. Jos ei vaadita, käyttäjä painaa ”OK”, jolloin huolto merkitään tehdyksi.

”Lisää huoltoraportti”-nappia painamalla käyttäjälle aukeaa tiedostonvalintaikkuna. Valittu tiedosto lisätään huollon huoltoraportiksi, se nimetään uudestaan kuvaamaan minkä huollon raportti se on, ja vaihdetaan ”Lisää huoltoraportti” teksti ”Katso huoltoraportti” tekstiin.

**Poikkeukset:** Sisäänkirjautuminen epäonnistuu: kirjautumistiedot ovat väärät.

Raportin lisääminen epäonnistuu: Valittu huolto on poistettu järjestelmästä tiedostoa valitessa.

Tehdyksi merkkäminen epäonnistuu: Valittu huolto on poistettu järjestelmästä sivulle tulon jälkeen.

**Lopputulos:** Käyttäjä on merkannut huollon tehdyksi ja lisännyt sille huoltoraportin. Tiedot on tallennettu järjestelmään.